Long-Term Security (Policy) Needs

DOE National Collaboratories Meeting
August 11th, 2004

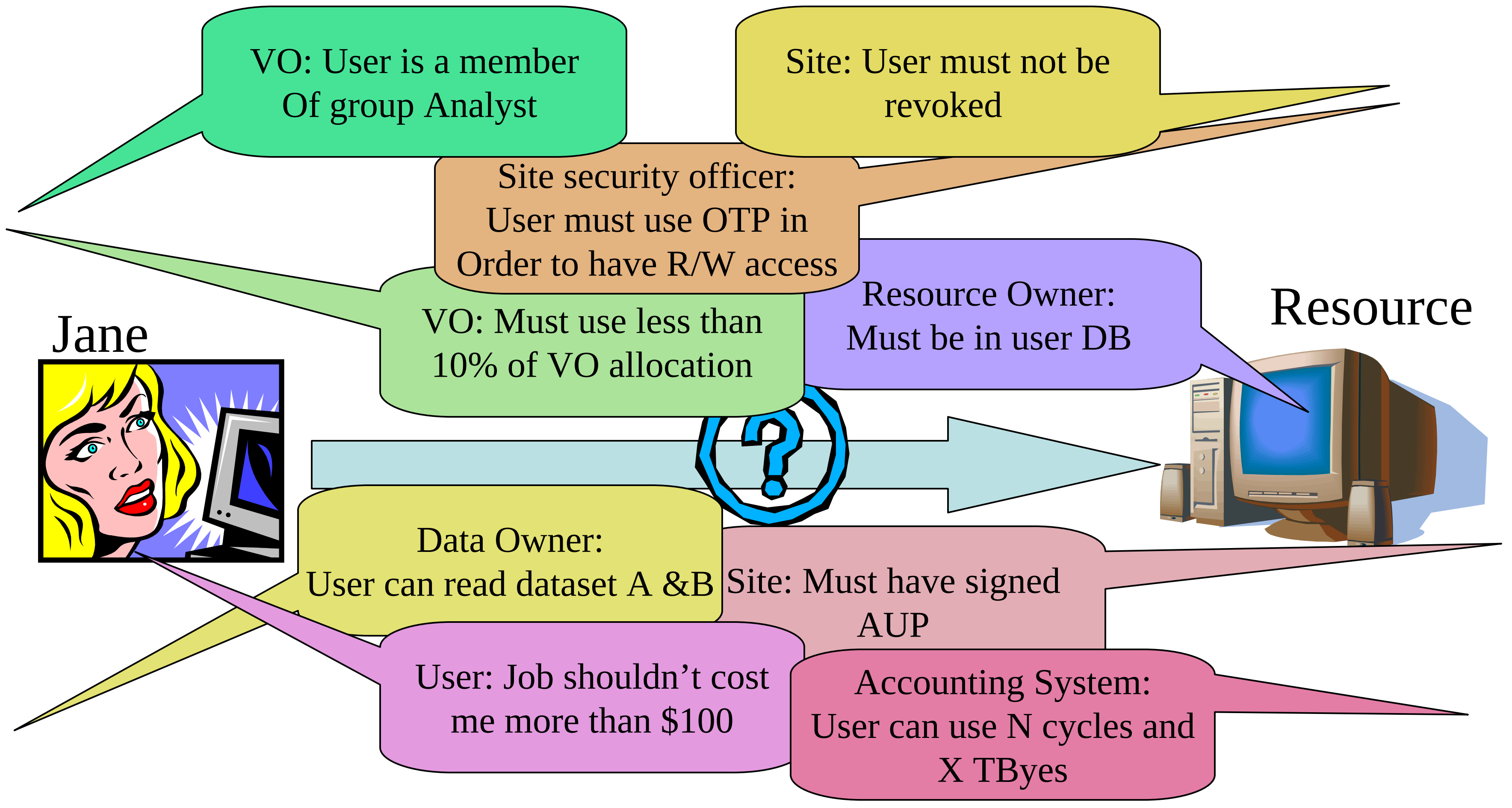
Von Welch, Frank Siebenlist, Sam Meder

Policy Needs of Collaboration

- A collaboration is defined by the combination of the polices of all it’s participants
  - Everyone wants their say in the final policy decisions
- Everyone does this today in their own site in their own way
- Today in the Grid we are dealing mainly with static policies, combined manually
  - Hard for one party to change their policy
  - Hard to do delegation
- How to allow combinations of policies from everyone in the way resource owner wants?

the globus alliance
www.globus.org

See Jane get authorized…


VO: User is a member Of group Analyst
Site: User must not be revoked
Site security officer: User must use OTP in Order to have R/W access
Resource Owner: Must be in user DB
Resource
Jane
VO: Must use less than 10% of VO allocation
Data Owner: User can read dataset A &B
Site: Must have signed AUP
User: Job shouldn’t cost me more than $100
Accounting System: User can use N cycles and X TByes

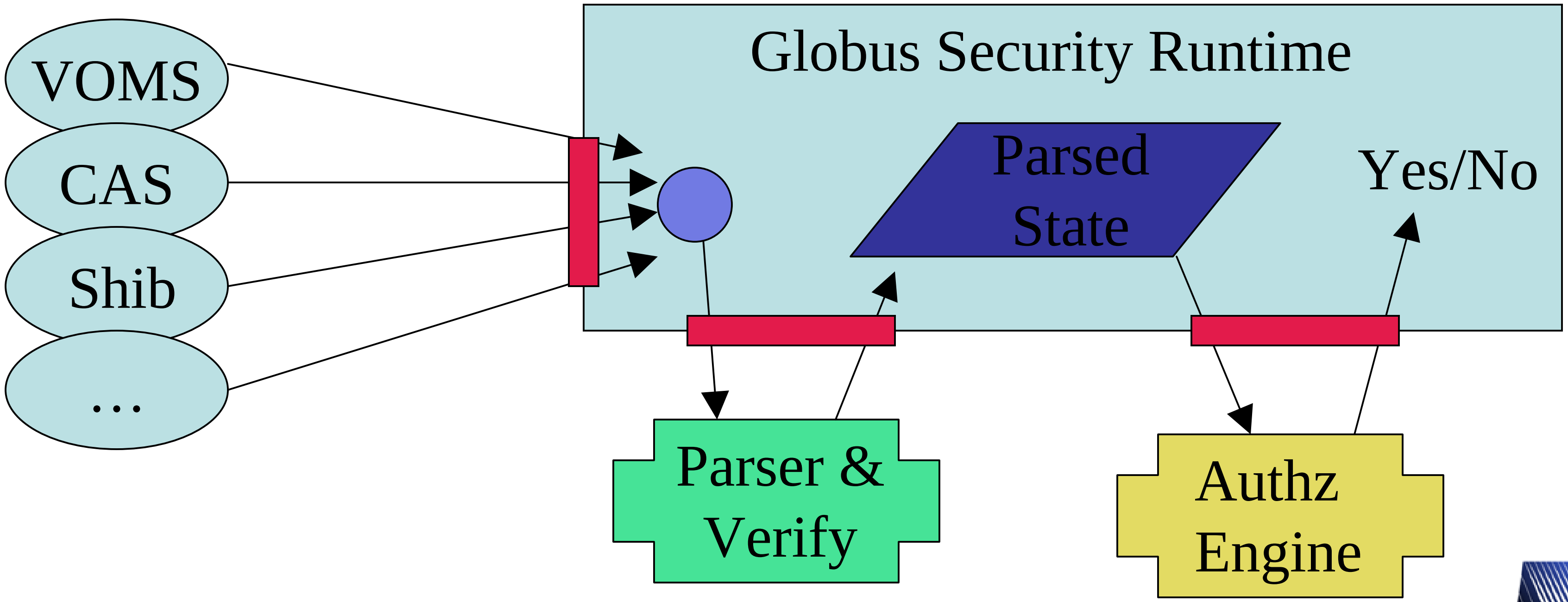
NCSA

Delegation is the key Feature for the users

- From Resource owner to VO
  - “This VO can use cycles, disk, instrument"
- From VO to user
  - Von is a member of group FOO and hence should be allowed to run jobs
- From user to user
  - Frank can read my data
- From user to service (job)
  - This service can use VO/my resources and access my data

Work in Progress: Grid Authorization Policy Architecture (GRAPA)

Key is structuring
Security system



the globus alliance
www.globus.org

Challenges

- Writing policies are hard
  - Policies are often “understood” and not written down
  - Writing a policy and verifying it is what you want is hard
- Matching of policy languages to other Grid languages
- As Miron says, when something goes wrong or is denied, figuring out why…
- How do we make credentials easy and trustworthy?

NCSA

Languages that “drive” the Grid applications

- •High level languages
  - –Agreement negotiation, job description, scheduling, etc.
  - –ws-agreement, jsdl, scheduling language efforts underway in GGF...
- •Policy languages
  - –Grid-mapfile, XACML, PERMIS, etc.
- •How to minimize and deal with mismatches between languages?

Language Mismatches

- Examples
  - File vs directory vs dataset
  - Service vs server vs cluster
- Consequences:
  - Any imperfect mapping of application language primitives to policy language equivalents always necessitates the granting of too many rights
  - In the case of compromise, the granting of too many rights leads to excessive exposure
  - When services work on behalf of other services and rights are delegated, the consequences become more prominent

the globus alliance
www.globus.org

A Slide for Miron

- Complicated system like this will need good error reporting/troubleshooting
- Lots of standards and software reuse are good, but result in lots of layers of software
- Problem is: Error == stack trace
  - Not useful to the end user
  - Each layer has some of needed context to understand error
  - User's read first two lines of an error message
- How to translate trace to something useful?
  - Today this is a manual, brittle process of identify stack and map to useful message

NCSA

DOE’s new realities…

- We’ve partly surrendered the desktop and servers
  - Not clear how much yet…
- Our desktops and servers will be compromised
  - How do we know when that happens?
- How to express levels of trust in different resources?
- Cleaning up compromised resources will be routine…
  - How to make this least painful?
- Unknowingly deploy compromised resources
  - How to limit the consequences?

Conclusions

- Need to find ways of writing, verifying, distributing, combining and debugging policies from multiple sources
- Writing policies is not common today
- These need to be coherent with the other languages we use so make evaluation possible
- Need agreement to requirements and semantics

the globus alliance
www.globus.org

Final Conclusion

- Firewalls will probably be a problem